Amendments to the Claims

Please amend Claims 1, 26, 49, and 50. The Claim Listing below will replace all prior versions of the claims in the application:

Claim Listing

1. (Currently Amended) A method for identifying a protected software comprising:

creating a superfingerprint for said protected software by:

executing said protected software at least once;

in each execution, using a supervising program, selecting specified portions of the executing image of at least one of said executing protected software and of results of executing said protected software;

in each execution, using the supervising program, performing computations on said selected portions to obtain a collection of fingerprints; and

combining, using the supervising program, said collections of fingerprints found in each execution into the superfingerprint of said protected software according to a combining rule[[.]];

at a later time, detecting execution of an unidentified software; and

using the superfingerprint of the protected software to identify the executing unidentified software, where the executing unidentified software is identified as the protected software using the superfingerprint even if the executing unidentified software and the protected software are not exactly the same.

2. (Previously Presented) The method of claim 1 wherein the protected software is executed a plurality of times in order to form the superfingerprint and the collection of fingerprints obtained during each execution are combined together according to the combining rule.

3. (Original) The method of claim 2 wherein the combining rule outputs only those fingerprints that are computed in more than a specified number of executions.

4. (Previously Presented) The method of claim 1 wherein the combining rule used to form the superfingerprint removes from the output those fingerprints that occur in more than a specified number of executions of specified other protected softwares.

5. (Previously Presented) The method of claim 4 wherein fingerprints are not removed if they belong to a same group of protected software as said protected software.

6. (Previously Presented) The method of claim 1 wherein a fingerprint belongs to the superfingerprints of several protected softwares.

7. (Previously Presented) The method of claim 6 further comprising:

storing in at least one data structure at least one fingerprint and means to identify said several protected softwares in whose superfingerprint said fingerprint is included.

8. (Previously Presented) The method of claim 7 wherein the means to identify is a bit vector data structure whose mth bit indicates whether the superfingerprint associated with the mth member of said several protected softwares includes said fingerprint.

9. (Previously Presented) The method of claim 7 wherein associated with each said fingerprint there are at least two numbers k1 and k2 where k2 is greater than or equal to k1 that indicate that the superfingerprints of protected softwares from k1 to k2 of said several protected softwares all include said fingerprint.

10. (Previously Presented) The method of claim 6 wherein said several protected softwares belong to a group of protected software.

11. (Previously Presented) The method of claim 1 wherein the fingerprints of various protected softwares are stored in a data structure to facilitate and accelerate retrieval of fingerprints and associated names of protected software.

12. (Previously Presented) The method of claim 1 wherein the executing protected software is partitioned into pages, said specified portions are selected from said pages and the computations produce a fingerprint for each portion.

13. (Previously Presented) The method of claim 1 wherein said specified portions are selected from the protected software stored in a memory of the device executing said protected software.

14. (Previously Presented) The method of claim 1 wherein said specified portions are selected from the protected software stored in secondary memory of the device executing said protected software.

15. (Original) The method of claim 1 wherein the specified portions are basic blocks of programs.

16. (Original) The method of claim 1 wherein the computation involves only parts of said selected portions.

17. (Original) The method of claim 16 wherein said involved parts are operation codes.

18. (Previously Presented) The method of claim 16 wherein said involved parts are information in an audio signal.

19. (Original) The method of claim 16 wherein said involved parts are information in a visual display.

20. (Previously Presented) The method of claim 1 wherein the selected portion concerns the interaction between at least one user and the execution of protected software.

21. (Original) The method of claim 1 wherein the input to the computation is a sequence.

22. (Original) The method of claim 1 wherein the computation is a hash function value of said portion.

23. (Original) The method of claim 22 wherein the hash function value is computed by polynomial fingerprinting.

24. (Original) The method of claim 1 wherein the computation is a computation on an audio signal.

25. (Original) The method of claim 1 wherein the computation is a computation on a video stream.

26. (Currently Amended) A method for identifying a first protected software comprising the steps of:

storing previously created superfingerprints for at least one protected software;

detecting execution of an unidentified software; and

using the superfingerprints of the protected software to identify the executing unidentified software, where the executing unidentified software is identified as the protected software using the superfingerprint even if the executing unidentified software and the protected software are not exactly the same by:

~~executing said first protected software at least once;~~

selecting by a supervising program specified portions of the executing image of at least one of said executing unidentified ~~protected~~ software and of the results of executing said ~~protected first~~ executing unidentified software on each execution;

performing by said supervising program specified computations on said selected portions to obtain a collection of fingerprints from the executing unidentified software;

comparing said collection of fingerprints from the executing unidentified software to said previously computed superfingerprint of ~~at least one second~~ the protected software to determine whether there is an approximate match; and

declaring said the executing unidentified software ~~first software~~ to be the same as said ~~second~~ the protected software if an approximate match is found.

27. (Previously Presented) The method of claim 26 wherein said specified portions of said executing protected software and of said results of executing said software, are stored in a memory of a device executing said protected software.

28. (Previously Presented) The method of claim 27 wherein said specified portions of at least one of said executing protected software and of said results of executing said protected

software, are selected from recently accessed portions of the protected software stored in the memory of the device executing said protected first software.

29. (Previously Presented) The method of claim 27 wherein said specified components of said executing protected software are selected from portions of said executing protected software stored in secondary storage.

30. (Previously Presented) The method of claim 26 wherein the portions of said executing protected software are selected while said protected software is sent from one device to another.

31. (Previously Presented) The method of claim 26 wherein said portions of the results of execution of said protected software are selected from the output of the device executing said protected software.

32. (Previously Presented) The method of claim 26 wherein said specified portions of at least one of said executing protected software and of the results of executing said protected software on a later execution are dependent on the results of an earlier approximate match.

33. (Previously Presented) The method of claim 26 wherein determining the approximate match comprises:

determining whether the amount of said commonality between said fingerprints of said first protected software and the fingerprints comprising said superfingerprint of said at least one second protected software exceeds a specified threshold in which case the first protected software is identified to be the same as the second protected software.

34. (Previously Presented) The method of claim 33 wherein said specified threshold is exceeded only if the amount of commonality between said fingerprints of said first protected software and the fingerprints comprising said superfingerprint of said second protected software exceed the commonality between said fingerprints of said first protected software and the fingerprints comprising the superfingerprint of a third protected software.

35. (Previously Presented) The method of claim 33 wherein the commonality between the fingerprints of said first protected software and said second protected software depends on the number of fingerprints that are the same in said two protected softwares with a weighting factor for each equal fingerprint.

36. (Previously Presented) The method of claim 35 wherein commonality further depends on the number of fingerprints that are different in said two protected softwares with a weighting factor for each unequal fingerprint.

37. (Previously Presented) The method of claim 35 wherein commonality further depends on the relative positions of the portions of protected software from which at least two fingerprints are computed.

38. (Original) The method of claim 26 wherein the specified computation involves only parts of said selected portions.

39. (Original) The method of claim 38 wherein said involved parts are operation codes.

40. (Original) The method of claim 38 wherein said involved parts are information in an audio signal.

41. (Original) The method of claim 38 wherein said involved parts are information in a visual display.

42. (Previously Presented) The method of claim 26 wherein the selected portion concerns the interaction between at least one user and the execution of protected software.

43. (Original) The method of claim 26 wherein the input to the computation is a sequence.

44. (Original) The method of claim 26 wherein the input to the comparison is a collection of fingerprints each having an associated weight.

45. (Original) The method of claim 26 wherein the computation is a hash function value of said portion.

46. (Original) The method of claim 45 wherein the hash function value is computed by polynomial fingerprinting.

47. (Original) The method of claim 26 wherein the computation is a computation on an audio signal.

48. (Original) The method of claim 26 wherein the computation is a computation on a video stream.

49. (Currently Amended) A method for identifying a protected software group of a first protected software comprising the steps of:

storing previously created superfingerprints for at least one protected software group;

~~executing said first protected software at least once;~~

detecting execution of an unidentified software; and

using one of the superfingerprints of the protected software to identify the executing unidentified software, where the executing unidentified software is identified as the protected software using the superfingerprint even if the executing unidentified software and the protected software are not exactly the same by:

selecting specified portions of the executing image of said executing ~~first protected~~ unidentified software and of the results of executing said ~~first protected~~ executing unidentified software on each execution;

performing specified computations on said selected portions to obtain a collection of fingerprints from the executing unidentified software;

comparing said collection of fingerprints to said previously computed superfingerprint of at least one second protected software group to determine whether there is an approximate match; and

declaring said ~~first protected~~ executing unidentified software to be a member of said second protected software group if an approximate match is found.

50. (Currently Amended) A method for identifying a protected software that is a member of a group of protected software comprising the steps of:

storing previously created superfingerprints for at least one protected software group and for members of that group;

~~executing said protected software at least once;~~

detecting execution of an unidentified software; and

using one of the superfingerprints of the protected software to identify the executing unidentified software, where the executing unidentified software is identified as the protected software using the superfingerprint even if the executing unidentified software and the protected software are not exactly the same by:

selecting specified portions of the executing image of said executing ~~protected~~ unidentified software and of the results of executing said ~~protected~~ executing unidentified software on each execution;

performing computations on said selected portions to obtain a collection of fingerprints from the executing unidentified software;

comparing said collection of fingerprints from the executing unidentified software to said previously computed superfingerprint of at least one ~~second~~ protected software group and the superfingerprints of the members of said ~~second~~ protected software group to determine whether there is an approximate match with said group and at least one of said superfingerprints of said members of said group; and

declaring said ~~protected~~ executing unidentified software to be the same as a particular member of said second protected software group if an approximate match is found.